1. Descreva o que faz o trecho de código a seguir:

```
String str1 = "abc" , str2 = "";
char[] letters = str1.toCharArray();
boolean isFirst = true;
for (char letter : letters) {
    str2 += isFirst ? ((int) letter) : "."+((int) letter);
    isFirst = false;
}
System.out.println(str2);
```

2. Explique como são aplicados os mecanismos de *Polimorfismo*, *Herança*, *Sobrescrita* e *Encapsulamento* nas classes ilustradas abaixo:

```
public class Colaborador {
    public String nome;
    private String matricula;
    public float salario;
    private boolean validaMatricula(String m){
        return m.length()==10;
    }
    public String getMatricula() {
        return matricula;
    }
    public void setMatricula(String matricula) {
        if(this.validaMatricula(matricula))
            this.matricula = matricula;
    }
    public double getBonificacao(){
        return this.salario * 0.15;
    }
}

class Gerente extends Colaborador{
    public String especialidade;
    public double getBonificacao(){
        return this.salario * 0.20;
    }
}
```

```
class ControleBonificacao{
    public double acumuladorBonoficacao;
    public void registrarBonoficacao(Colaborador c){
        if(c != null)
            this.acumuladorBonoficacao += c.getBonificacao();
    }
}
```

3. Escreva uma classe que represente lembretes. Um lembrete tem como atributos *descrição, dia, mês, ano*. Represente a classe seguindo as seguintes restrições:

a) Os atributos devem estar encapsulados, portanto crie os métodos *getters* e *setters*.

b) Especificar o construtor padrão e mais um que inicialize os quatro atríbutos;

c) Um método que permita verificar se dois lembretes são iguais. Dois lembretes são iguais se tiverem a mesma *descrição*, *dia, mês, ano*. A assinatura deste método deve ser:

public boolean equals(Lembrete lembrete)

d) Método *toString* que retorna uma String no seguinte formato:

"Descrição – dd/mm/aaaa"

4. Fazer um programa para simular o registro de dados de colaboradores de uma empresa. Para tal, crie uma uma classe *Colaborador* com os seguintes dados:

- *CPF: java.lang.String*

- *Nome: java.lang.String*

- *Sexo (M/F): char*

- *Salário Bruto: float/double*

- *Data de Admissão: java.util.GregorianCalendar*

São requisitos da classe da funcionário:

- Fornecer um método *validaCPF( )* para validar o CPF do objeto corrente. Considere um CPF válido, se o mesmo tem 14 caracteres;

- Fornecer um método *validaDataAdmissao( )* para validar a Data de Admissão do objeto corrente. Considere que a empresa foi fundada em 1990;

- Forneça um método *trabalhaMaisTempo( )* que recebe um objeto funcionário e retorna *false* se este trabalhar a mais tempo que o objeto corrente (*this*) e *true* caso contrário.

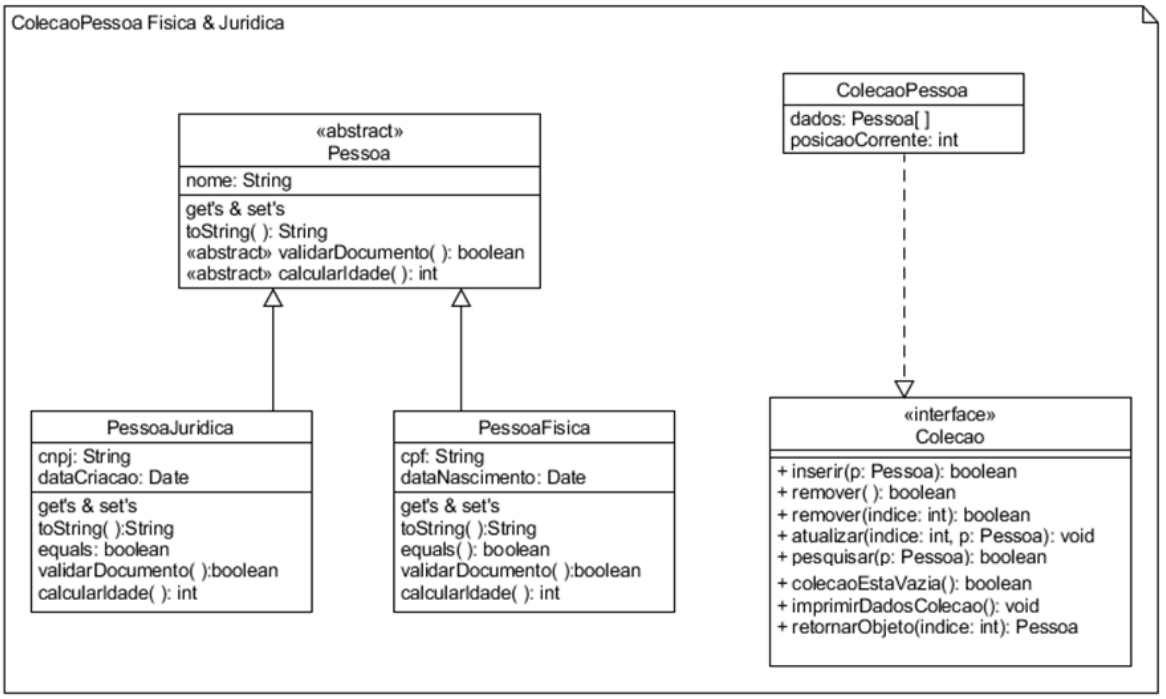
Implemente também uma classe *OperacaoColaborador* onde:

- Escreva um método estático a qual recebe um *array* de *Colaborador* e diz quais objetos *Colaborador* possuem dados válidos/inválidos (cpf e data de admissão);

- Escreva um método estático a qual recebe um *array* de *Colaborador* e uma *string* contendo um *cpf*. Caso o *cpf* seja válido, imprimir se há no vetor recebido um colaborador com o *cpf* também recebido;

- Escreva um método estático a qual recebe um *array* de *Colaborador* e imprime o nome e data de admissão do funcionário que trabalha a mais tempo na empresa. Obs: Considerar apenas anos.

5. Implementadas as seguintes classes e teste-as em uma classe *App*.



Restrições:

1. CPF é válido se a String tiver tamanho 14; Ex: XXX.XXX.XXX-ZZ
2. CNPJ é válido se a String tiver tamanho 18. Ex: XX.XXX.XXX/YYYY-ZZ;
3. Calculo da idade é o número de anos resultante da diferença entre a data atual e data de origem (nascimento ou criação). Usar *GreogiranCalendar* ao invés de Date;
4. Duas pessoas físicas são iguais se tiverem o mesmo cpf;
5. Duas pessoas jurídicas são iguais se tiverem o mesmo cnpj;
6. Nas classes a serem implementadas, todos os atributos devem ser *private;*
7. O método *pesquisar* recebe um objeto *p* e retorna *true* ou *false*, caso *p* esteja na coleção;
8. O vetor *dados* deve ter tamanho 100.